UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/679,498	10/07/2003	Burkhard K. Neidecker-Lutz	13909-128001 / 2003P00162	8438

32864 7590 01/07/2008

FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
DWIVEDI, MAHESH H	

ART UNIT	PAPER NUMBER
2168	

MAIL DATE	DELIVERY MODE
01/07/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/679,498	NEIDECKER-LUTZ, BURKHARD K.	
	Examiner	Art Unit	
	Mahesh H. Dwivedi	2168	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *25 October 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-14 and 21-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-14 and 21-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *07 October 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

Remarks

1. Receipt of Applicant's Amendment, filed on 10/25/2007, is acknowledged. The amendment includes the cancellation of claims 15-20, the amending of claims 1, 9, 14, and 24, and the addition of claims 25-27

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claim 22 recites the limitation "sorted memory buffer" in page 05. There is insufficient antecedent basis for this limitation in the claim, as no memory buffer is recited in independent claim 9.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1-14, and 21-24 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Amor et al.** (U.S. Patent 6,546,382) and **Microsoft** (Article entitled "TOP N Clause vs. SET ROWCOUNT, dated 05/21/2001).

6. Regarding claim 1, **Amor** teaches a data store query system comprising:

A) a data store that includes a collection of records; (Column 3, lines 26-31);

B) a constant-sized sorted result buffer (Column 3, lines 31-47); and

C) a query interface operable to: receive a limit and order query that includes both of an order criteria and a limit criteria (Column 1, lines 49-53, Column 2, lines 51-58);

D) the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query (Column 1, lines 54-59); and

E) to output the sorted result buffer as the result set of records (Column 5, lines 13-15);

F) fill the sorted result buffer with a first N number of records from the data store (Column 3, lines 40-44);

I) iteratively modify the query to include a where criteria that compares remaining records in the data store (Column 2, lines 51-58, Column 4, lines 32-44);

J) iteratively compare the remaining records in the data store against the record in the sorted result buffer based upon the order criteria and the where criteria (Column 4, lines 12-16, lines 32-44);

K) iteratively replace the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer (Column 4, lines 32-44); and

L) output the sorted result buffer as the result set of records (Column 5, lines 9-15)

The examiner notes that **Amor** teaches **"a data store that includes a collection of records"** as "The process is performed when a database system detects a Row Restricted Orderby Subquery and scans data from the table that contains the table referenced by the ORDER BY clause. The process is based on the formation of one or more TOP N subsets" (Column 3, lines 26-31). The examiner further notes that **Amor** teaches **"a constant-sized sorted result buffer"** as "A TOP N subset is a subset of rows that the process determines cannot be excluded from the TOP N rows in the order requested by a query. That is, a TOP N subset contains rows that are candidates for the TOP N rows in order" (Column 3, lines 31-34) and "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"a query interface operable to: receive a limit and order query that includes both of an order criteria and a limit criteria"** as "Rows may be returned in ascending or descending order. The default is ascending. The return order may be specified using the keyword ASC for ascending or DESC for descending" (Column 1, lines 49-53) and "According to an embodiment of the present invention, a

TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58). The examiner further notes that **Amor** teaches **"the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query"** as "A user that requests ordered data may desire only the TOP N rows in order. The term "TOP N" refers to the first N data items in an ordered set of data items. For example, the first 10 rows from payroll in ascending order based on salary. An operation or process that returns the TOP N data items based on an order is referred to as a TOP N operation" (Column 1, lines 54-59). The examiner further notes that **Amor** teaches **"to output the sorted result buffer as the result set of records"** as "The execution of the steps ends, and the result set is returned in TOP N order" (Column 5, lines 13-15). The examiner further notes that **Amor** teaches **"fill the sorted result buffer with a first N number of records from the data store"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"iteratively modify the query to include a where criteria that compares remaining records in the data store"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58) and "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the

entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"iteratively compare the remaining records in the data store against the record in the sorted result buffer based upon the order criteria and the where criteria"** as "At step 126, the sort value of the scanned row is compared with the threshold to determine whether the row belongs in the current TOP N subset. In this example, the scanned row has a sort value of 75000, which is greater than the threshold. Therefore, the row belongs to the TOP N subset" (Column 4, lines 12-16). The examiner further notes that **Amor** teaches **"to iteratively replace the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"output the sorted result buffer as the result set of records"** as "The execution of the steps ends, and the result set is returned in TOP N order" (Column 5, lines 13-15).

Amor does not explicitly teach:

G) to iteratively order the sorted result buffer based upon the order criteria;

H) iteratively identify the last record of the sorted result buffer;

I) to the identified last record based on a sort order determined by the order criteria;

J & K) the identified last record

Microsoft, however, teaches **"to iteratively order the sorted result buffer based upon the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is

nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), "**iteratively identify the last record of the sorted result buffer**" as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), "**to the identified last record based on a sort order determined by the order criteria**" as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), and "**the identified last record**" as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1)

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 2, **Amor** further teaches a data store query system comprising:
A) wherein the data store is a database or a fast cache (Column 1, lines 60-67).

The examiner notes that **Amor** teaches **"wherein the data store is a database or a fast cache"** as "To retrieve the top 10 salary values stored in the salary column of the payroll table, a user process issues a query to the database system that contains payroll. To the get rows with the top 10 salary values, the user issues the query B2 to the database system. The database system returns to the user all the rows from payroll in an order according to the values in salary. The user then retains the first 10 rows received, and discards the rest" (Column 1, lines 60-67).

Regarding claim 3, **Amor** further teaches a data store query system comprising:
A) wherein the collection of records further comprises a table having an attribute (Column 1, lines 60-67, Column 2, lines 51-58); and
B) wherein the query interface is operable to receive the limit and order query placing order constraints on the attribute (Column 1, lines 60-67, Column 2, lines 51-58).

The examiner notes that **Amor** teaches **"wherein the collection of records further comprises a table having an attribute"** as "To retrieve the top 10 salary values stored in the salary column of the payroll table, a user process issues a query to the database system that contains payroll. To the get rows with the top 10 salary values, the user issues the query B2 to the database system. The database system returns to the user all the rows from payroll in an order according to the values in salary. The user then retains the first 10 rows received, and discards the rest" (Column 1, lines 60-67) and "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58). The examiner further notes that **Amor** teaches **"wherein the query interface is operable to receive the limit and order query placing order constraints on the attribute"** as "To retrieve the top 10 salary values stored in the salary column of the payroll table, a user process issues a query to the database system that contains payroll. To the get rows with the top 10 salary values, the user issues the query B2 to the database system. The database system returns to the user all the rows from payroll in an order according to the values in salary. The user then retains the first 10 rows received, and discards the rest" (Column 1, lines 60-67) and "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58).

Regarding claim 4, **Amor** further teaches a data store query system comprising: A) wherein the query interface creates a revised sorted result buffer in response to a modification of the limit and order query, the modification being made during a pause in execution of the limit and order query (Column 4, lines 32-44).

The examiner notes that **Amor** teaches **"wherein the query interface creates a revised sorted result buffer in response to a modification of the limit and order query, the modification being made during a pause in execution of the limit and order query"** as "Adding a new member to and removing a new member from the current TOP N subset may change the threshold used to determine whether a particular scanned row qualifies for the TOP N subset. For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44).

Regarding claim 5, **Amor** further teaches a data store query system comprising: A) wherein the sorted result buffer is stored in random access memory (Column 3, lines 48-55).

The examiner notes that **Amor** teaches **"wherein the collection of records further comprises a table having an attribute"** as "One of the TOP N subsets is stored in one or more buffers ("memory buffers") in volatile memory. This TOP N subset is referred to as the current TOP N subset. Initially, the initial TOP N subset is the current TOP N subset" (Column 3, lines 51-55).

Regarding claim 6, **Amor** further teaches a data store query system comprising:

A) wherein the query interface is operable to receive the limit and order query formulated using standard query language (SQL) (Column 1, lines 13-15, lines 34-35).

The examiner notes that **Amor** teaches **"wherein the query interface is operable to receive the limit and order query formulated using standard query language (SQL)"** as "Users of database systems retrieve data through the use of queries. A query is a request for data. Typically, queries must conform to the rules of a particular query language, such as the ANSI Structured Query Language (SQL)" (Column 1, lines 13-15) and "When a database system executes an SQL query, the database system returns results in the form of a set of rows" (Column 1, lines 34-35).

Regarding claim 7, **Amor** further teaches a data store query system comprising: A) wherein the query interface is operable to receive the limit and order query that requests the first or last N records satisfying the query (Column 1, lines 49-52).

The examiner notes that **Amor** teaches **"wherein the query interface is operable to receive the limit and order query that requests the first or last N records satisfying the query"** as "Rows may be returned in ascending or descending order. The default is ascending. The return order may be specified using the keyword ASC for ascending or DESC for descending" (Column 1, lines 49-52).

Regarding claim 8, **Amor** further teaches a data store query system comprising: A) wherein the query interface is operable to identify data in the data store that satisfies the limit and order query using the sorted result buffer by iteratively reformulating the limit and order query until the sorted result buffer contains the satisfying limit and order query (Column 4, lines 32-44).

The examiner notes that **Amor** teaches **"wherein the query interface is operable to identify data in the data store that satisfies the limit and order query using the sorted result buffer by iteratively reformulating the limit and order query until the sorted result buffer contains the satisfying limit and order query"** as "Adding a new member to and removing a new member from the current TOP N

subset may change the threshold used to determine whether a particular scanned row qualifies for the TOP N subset. For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44).

Regarding claim 9, **Amor** teaches a method comprising:

A) receiving a limit and order query that includes both of an order criteria and a limit criteria (Column 2, lines 51-58);

B) the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query (Column 1, lines 54-59);

C) filling a constant-sized sorted result buffer with a first N number of records from a data store (Column 3, lines 40-44);

F) iteratively modifying the query to include a where criteria that compares remaining records in the data store (Column 2, lines 51-58, Column 4, lines 32-44);

G) iteratively comparing the remaining records in the data store against the record in the sorted result buffer based upon the order criteria and the where criteria (Column 4, lines 12-16, lines 32-44);

H) iteratively replacing the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer (Column 4, lines 32-44); and

I) outputting the sorted result buffer as the result set of records (Column 5, lines 13-15).

The examiner notes that **Amor** teaches **"receiving a limit and order query that includes both of an order criteria and a limit criteria"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the

result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58). The examiner further notes that **Amor** teaches **"the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query"** as "A user that requests ordered data may desire only the TOP N rows in order. The term "TOP N" refers to the first N data items in an ordered set of data items. For example, the first 10 rows from payroll in ascending order based on salary. An operation or process that returns the TOP N data items based on an order is referred to as a TOP N operation" (Column 1, lines 54-59). The examiner further notes that **Amor** teaches **"filling a constant-sized sorted result buffer with a first N number of records from a data store"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"iteratively modifying the query to include a where criteria that compares remaining records in the data store"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58) and "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"iteratively comparing the remaining records in the data store against the**

record in the sorted result buffer based upon the order criteria and the where criteria" as "At step 126, the sort value of the scanned row is compared with the threshold to determine whether the row belongs in the current TOP N subset. In this example, the scanned row has a sort value of 75000, which is greater than the threshold. Therefore, the row belongs to the TOP N subset" (Column 4, lines 12-16). The examiner further notes that **Amor** teaches **"iteratively replacing the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"outputting the sorted result buffer as the result set of records"** as "The execution of the steps ends, and the result set is returned in TOP N order" (Column 5, lines 13-15).

Amor does not explicitly teach:

D) iteratively ordering the sorted result buffer based upon the order criteria;

E) iteratively identifying the last record of the sorted result buffer;

F) to the identified last record based on a sort order determined by the order criteria;

G & H) identified last record

Microsoft, however, teaches **"iteratively ordering the sorted result buffer based upon the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"iteratively identifying the last record of the sorted result buffer"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in

which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"to the identified last record based on a sort order determined by the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), and **"identified last record"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1).

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 10, **Amor** further teaches a method comprising:
A) wherein the limit and order query is formulated using standard query language (SQL) (Column 1, lines 13-15, lines 34-35).

The examiner notes that **Amor** teaches **"wherein the limit and order query is formulated using standard query language (SQL)"** as "Users of database systems retrieve data through the use of queries. A query is a request for data. Typically, queries must conform to the rules of a particular query language, such as the ANSI Structured Query Language (SQL)" (Column 1, lines 13-15) and "When a database system executes an SQL query, the database system returns results in the form of a set of rows" (Column 1, lines 34-35).

Regarding claim 11, **Amor** further teaches a method comprising:

A) wherein filling the sorted result buffer with the first N number of records comprises scanning the data store without consideration of the order criteria to identify records otherwise satisfying the limit and order query (Column 3, lines 40-44); and
B) placing identified records into the sorted result buffer until the sorted result buffer includes the maximum number of records specified by the limit criteria (Column 3, lines 40-44).

The examiner notes that **Amor** teaches **"wherein filling the sorted result buffer with the first N number of records comprises scanning the data store without consideration of the order criteria to identify records otherwise satisfying the limit and order query"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"placing identified records into the sorted result buffer until the sorted result buffer includes the maximum number of records specified by the limit criteria"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44).

Regarding claim 12, **Amor** further teaches a data store query system comprising:
A) wherein the limit and order query requests the first N records satisfying the order criteria (Column 1, lines 49-52).

The examiner notes that **Amor** teaches **"wherein the limit and order query requests the first N records satisfying the order criteria"** as "Rows may be returned in ascending or descending order. The default is ascending. The return order may be specified using the keyword ASC for ascending or DESC for descending" (Column 1, lines 49-52).

Regarding claim 13, **Amor** further teaches a data store query system comprising:
A) wherein the limit and order query requests the last N records satisfying the order criteria (Column 1, lines 49-52).

The examiner notes that **Amor** teaches **"wherein the limit and order query requests the last N records satisfying the order criteria"** as "Rows may be returned in ascending or descending order. The default is ascending. The return order may be specified using the keyword ASC for ascending or DESC for descending" (Column 1, lines 49-52).

Regarding claim 14, **Amor** teaches an apparatus comprising:
A) a first code segment for receiving a limit and order query that includes both of an order criteria and a limit criteria (Column 2, lines 51-58);
B) the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query (Column 1, lines 54-59);
C) a second code segment for filling a constant-sized sorted result buffer with a first N number of records from a data store (Column 3, lines 40-44);
F) a fifth code segment for iteratively modifying the query to include a where criteria that compares remaining records in the data store (Column 2, lines 51-58, Column 4, lines 32-44);
G) a sixth code segment for iteratively comparing remaining records in the data store against a Nth record in the sorted result buffer based upon the order criteria and the where criteria (Column 4, lines 12-16, lines 32-44);
H) a seventh code segment for iteratively replacing the Nth record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the Nth record in the sorted result buffer (Column 4, lines 32-44); and
I) an eighth code segment for outputting the sorted result buffer as the result set of records (Column 5, lines 13-15).

The examiner notes that **Amor** teaches **"a first code segment for receiving a limit and order query that includes both of an order criteria and a limit criteria"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58). The examiner further notes that **Amor** teaches **"the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query"** as "A user that requests ordered data may desire only the TOP N rows in order. The term "TOP N" refers to the first N data items in an ordered set of data items. For example, the first 10 rows from payroll in ascending order based on salary. An operation or process that returns the TOP N data items based on an order is referred to as a TOP N operation" (Column 1, lines 54-59). The examiner further notes that **Amor** teaches **"a second code segment for filling a constant-sized sorted result buffer with a first N number of records from a data store"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"a fifth code segment for iteratively modifying the query to include a where criteria that compares remaining records in the data store"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58) and "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value

of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"a sixth code segment for iteratively comparing remaining records in the data store against a Nth record in the sorted result buffer based upon the order criteria and the where criteria"** as "At step 126, the sort value of the scanned row is compared with the threshold to determine whether the row belongs in the current TOP N subset. In this example, the scanned row has a sort value of 75000, which is greater than the threshold. Therefore, the row belongs to the TOP N subset" (Column 4, lines 12-16). The examiner further notes that **Amor** teaches **"a seventh code segment for iteratively replacing the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"an eighth code segment for outputting the sorted result buffer as the result set of records"** as "The execution of the steps ends, and the result set is returned in TOP N order" (Column 5, lines 13-15).

Amor does not explicitly teach:

D) a third code segment for iteratively ordering the sorted result buffer based upon the order criteria;

E) a fourth code segment for iteratively identifying the last record of the sorted result buffer;

F) to the identified last record based on a sort order determined by the order criteria;

H) identified last record

Microsoft, however, teaches **"a third code segment for iteratively ordering the sorted result buffer based upon the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"a fourth code segment for iteratively identifying the last record of the sorted result buffer"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"to the identified last record based on a sort order determined by the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), and **"identified last record"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1).

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 21, **Amor** further teaches a data store query system comprising:
A) wherein the size of the constant-sized sorted result buffer is based on the limit criteria (Column 3, lines 40-44).

The examiner notes that **Amor** teaches **"wherein the size of the constant-sized sorted result buffer is based on the limit criteria"** as "the initial TOP N set is

formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44).

Regarding claim 22, **Amor** does not explicitly teach a method comprising:
A) wherein iteratively replacing the Nth record in the sorted result buffer further comprises iteratively replacing the last record in the sorted memory buffer.

Microsoft, however, teaches **"wherein iteratively replacing the Nth record in the sorted result buffer further comprises iteratively replacing the last record in the sorted memory buffer"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1).

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 23, **Amor** does not explicitly teach a method comprising:
A) wherein iteratively ordering the sorted result buffer further comprises ordering the sorted result buffer for each iterative replacement of the Nth record.

Microsoft, however, teaches **"wherein iteratively ordering the sorted result buffer further comprises ordering the sorted result buffer for each iterative replacement of the Nth record"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is

nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1).

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 24, **Amor** teaches a method comprising:

A) receiving a standard query language (SQL) formatted limit and order query that includes both of an order criteria and a limit criteria (Column 2, lines 51-58);

B) the limit and order criteria requesting the first or last N records satisfying the order criteria (Column 1, lines 54-59); and

C) the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query (Column 1, lines 54-59);

D) filling a constant-sized sorted result buffer with a first N number of records from a data store of a customer relationship management system, an enterprise resource planning system or a supply chain management system (Column 3, lines 40-44);

E) the data store comprising a database or a fast cache (Column 1, lines 60-67);

F) wherein filling the sorted result buffer with the first number of records from the data store further comprises: scanning the data store without consideration of the order criteria to identify records otherwise satisfying the limit and order query (Column 3, lines 40-44); and

G) placing identified records into the sorted result buffer until the sorted result buffer includes the maximum number of records specified by the limit criteria (Column 3, lines 40-44);

J) <u>iteratively modifying the query to include a where criteria that compares remaining records in the data store</u> (Column 2, lines 51-58, Column 4, lines 32-44);

K) iteratively comparing the remaining records in the data store against a Nth record in the sorted result buffer based upon the order criteria and the where criteria (Column 4, lines 12-16, lines 32-44);
L) iteratively replacing the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer (Column 4, lines 32-44); and
M) outputting the sorted result buffer as the result set of records (Column 5, lines 13-15).

The examiner notes that **Amor** teaches **"receiving a standard query language (SQL) formatted limit and order query that includes both of an order criteria and a limit criteria"** as "According to an embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58). The examiner further notes that **Amor** teaches **"the limit and order criteria requesting the first or last N records satisfying the order criteria"** as "A user that requests ordered data may desire only the TOP N rows in order. The term "TOP N" refers to the first N data items in an ordered set of data items. For example, the first 10 rows from payroll in ascending order based on salary. An operation or process that returns the TOP N data items based on an order is referred to as a TOP N operation" (Column 1, lines 54-59). The examiner further notes that **Amor** teaches **"the limit criteria specifying a maximum number N of records for a result set of records satisfying the limit and order query"** as "A user that requests ordered data may desire only the TOP N rows in order. The term "TOP N" refers to the first N data items in an ordered set of data items. For example, the first 10 rows from payroll in ascending order based on salary. An operation or process that returns the TOP N data items based on an order is referred to as a TOP N operation" (Column 1, lines 54-59). The examiner further notes that **Amor** teaches **"filling a constant-sized sorted result buffer with a first N number of**

records from a data store of a customer relationship management system, an enterprise resource planning system or a supply chain management system" as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"the data store comprising a database or a fast cache"** as "To retrieve the top 10 salary values stored in the salary column of the payroll table, a user process issues a query to the database system that contains payroll. To the get rows with the top 10 salary values, the user issues the query B2 to the database system. The database system returns to the user all the rows from payroll in an order according to the values in salary. The user then retains the first 10 rows received, and discards the rest" (Column 1, lines 60-67). The examiner further notes that **Amor** teaches **"wherein filling the sorted result buffer with the first number of records from the data store further comprises: scanning the data store without consideration of the order criteria to identify records otherwise satisfying the limit and order query"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"placing identified records into the sorted result buffer until the sorted result buffer includes the maximum number of records specified by the limit criteria"** as "the initial TOP N set is formed. The initial TOP N set includes the first N rows scanned. The initial N rows scanned are in the TOP N subset because, at least initially, they may all be TOP N rows. In this example, the first 10 rows scanned are from the payroll table, and thus the initial TOP N subset is formed" (Column 3, lines 40-44). The examiner further notes that **Amor** teaches **"iteratively modifying the query to include a where criteria that compares remaining records in the data store"** as "According to an

embodiment of the present invention, a TOP N operation is performed through the use of a subquery that includes an ORDER BY clause and a restriction that references the result set of the subquery. The following query EX is provided as an example: SELECT salary FROM (SELECT salary FROM payroll ORDER BY salary) WHERE rownum<10" (Column 2, lines 51-58) and "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"iteratively comparing the remaining records in the data store against a Nth record in the sorted result buffer based upon the order criteria and the where criteria"** as "At step 126, the sort value of the scanned row is compared with the threshold to determine whether the row belongs in the current TOP N subset. In this example, the scanned row has a sort value of 75000, which is greater than the threshold. Therefore, the row belongs to the TOP N subset" (Column 4, lines 12-16). The examiner further notes that **Amor** teaches **"iteratively replacing the record in the sorted result buffer with a remaining record in the data store based upon iteratively comparing remaining records in the data store against the record in the sorted result buffer"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner further notes that **Amor** teaches **"outputting the sorted result buffer as the result set of records"**

as "The execution of the steps ends, and the result set is returned in TOP N order" (Column 5, lines 13-15).

Amor does not explicitly teach:

H) iteratively ordering the sorted result buffer based upon the order criteria;

I) iteratively identifying the last record of the sorted result buffer;

J) to the identified last record based on a sort order determined by the order criteria;

L) identified last record.

Microsoft, however, teaches "iteratively **ordering the sorted result buffer based upon the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"iteratively identifying the last record of the sorted result buffer"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), **"to the identified last record based on a sort order determined by the order criteria"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1), and **"identified last record"** as "With unsorted input, the TOP N operator uses a small internal sorted temporary table in which it replaces only the last row. If the input is nearly sorted, the TOP N engine must delete or insert the last row only a few times" (Page 1).

The examiner further notes that it is common knowledge that the TOP N function in SQL is an iterative sort function.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Microsoft's** would have allowed **Amor's** to provide a method for faster sorting of desired index records, as noted by **Microsoft** (Page 1).

Regarding claim 25, **Amor** further teaches a method comprising:

A) wherein execution of the query is paused when the query is modified (Column 4, lines 32-44).

The examiner notes that **Amor** teaches **"wherein execution of the query is paused when the query is modified"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44).

Regarding claim 26, **Amor** further teaches a method comprising:

A) wherein execution of the query is resumed at the current table location after the query is modified (Column 4, lines 32-44).

The examiner notes that **Amor** teaches **"wherein execution of the query is resumed at the current table location after the query is modified"** as "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44).

Regarding claim 27, **Amor** further teaches a method comprising:

A) wherein the sort order is a "less than" sort order (Column 1, lines 49-53).

The examiner notes that **Amor** teaches **"wherein the sort order is a "less than" sort order"** as "Rows may be returned in ascending or descending order. The

default is ascending. The return order may be specified using the keyword ASC for ascending or DESC for descending" (Column 1, lines 49-53).

Response to Arguments

7. Applicant's arguments filed on 10/25/2007 have been fully considered but they are not persuasive.

Applicants argue on page 09 that **"Logically, it must then follow that Amor also does not disclose the newly claimed features that i) the query is iteratively modified to include a where criteria that compares remaining records in the data store to the identified last record based on a sort order determined by the order criteria...and the where criteria"**. However, the examiner wishes to refer to Column 04 of **Amor** which states "For example, in the current illustration, the member of the TOP N subset that was removed had a sort value of 50000, the value upon which the current threshold was based. After removing that row from the TOP N set and adding the current row, the lowest sort value of any member of the current TOP N subset is 60000. At step 148, the entry threshold is recalculated to the lowest sort value of the rows that belong to the current TOP N subset. In the current illustration, the entry threshold is adjusted to 60000" (Column 4, lines 32-44). The examiner wishes to state that by continuing to readjust the threshold value, the where criteria of the top n query is modified because the threshold value is modified. Moreover, because the where criteria after each iteration is different from the previous iteration (where the threshold has changed), a new where criteria is included after each iteration. Furthermore, the order criteria is already established in the initial query, (i.e. either ascending or descending). Therefore, the combination of **Amor** and **Microsoft** (teaching comparing to the last record) teaches the aforementioned limitations.

Applicants argue on page 09 that **"the Microsoft article also fails to cure Amor's deficiencies...nothing in this short reference is seen to describe the iterative modification of a query to include a where criteria that compares remaining records in the data store to an identified last record based on a sort order determined by the order criteria. Absent such a description, the Microsoft**

Article cannot also been seen to iteratively compare remaining records based on the where criteria.". However, the examiner wishes to state that **Amor** teaches the comparison based on a where criteria. Moreover, the examiner wishes to state that applicants should know that common SQL sort functions such as top N contain comparative functions; and that it is implicit that such sort functions contain comparative operators (i.e. one cannot sort an unsorted array without performing comparative operations).

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

Article entitled "Oracle9i: Data Cartridge Developer's Guide" by **Gietz et al.**, (June 2001). The subject matter disclosed therein is pertinent to that of claims 1-14, and 21-27 (Pipelining).

U.S. Patent 5,671,403 issued to **Shekita et al.** on 23 September 1997. The subject matter disclosed therein is pertinent to that of claims 1-14, and 21-27 (e.g., methods to iteratively attain query results).

Article entitled "Single Buffered Histogram Sort" by **McCoskey**, (04 February 1999). The subject matter disclosed therein is pertinent to that of claims 1-14, and 21-27 (Pipelining).

U.S. Patent 5,974408 issued to **Cohen et al.** on 26 October 1999. The subject matter disclosed therein is pertinent to that of claims 1-14, and 21-27 (e.g., methods to iteratively attain query results).

U.S. PGPUB 2003/0233340 issued to **Flasza et al.** on 18 December 2003. The subject matter disclosed therein is pertinent to that of claims 1-14, and 21-27 (e.g., methods to iteratively attain query results).

9. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

10. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

January 02, 2008

TIM VO
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100